Third Interop–Vlab.It Workshop on Enterprise Interoperability

Scientific Research and Opportunities for Business Realities

Co-located with itAIS 2010
Università degli Studi di Napoli "Parthenope"

Napoli, October 9, 2010

Organized by: Laboratory for Enterprise Knowledge and Systems

In partnership with: AICA Associazione Italiana per l'Informatica ed il Calcolo Automatico *Sezione di Roma*

Istituto di Analisi dei Sistemi ed Informatica · Università degli Studi di Milano · LUISS · ENEA · TXT e-solutions

Third Interop–Vlab.It Workshop

Scientific Committee

Paola Velardi	Università “La Sapienza”di Roma
Alessandro D'Atri	CERSI LUISS Guido Carli
Michele Missikoff	IASI-CNR
Elaheh Pourabbas	IASI-CNR

Organization Committee

Lada Vetrini	IASI-CNR
Fabrizio Smith	IASI-CNR, Università de L’Aquila
Paolo Spagnoletti	CERSI LUISS Guido Carli
Stefano Za	CERSI LUISS Guido Carli
Rocco Agrifoglio	Università degli Studi di Napoli “Parthenope”

Preface

This volume collects the papers of the *Third Interop-Vlab.IT Workshop on Enterprise Interoperability* held in Naples, on the 9th October 2010. The workshop is promoted and supported by the "*Polo di ricerca scientifica e tecnologica sull'interoperabilità – Interop-Vlab.IT*" a scientific, cultural and non-profit association aimed at promoting research and initiatives on enterprise interoperability, from the technological, to the methodological, organizational, and cultural perspectives.

Interoperability mainly concerns enterprises, public administration, business processes, organizations, people, and software applications. The preconditions to develop interoperability solutions are:

- cross-fertilization between different research communities, mainly, enterprise modelling, ontologies, software architectures and platforms;
- definition and development of methods supporting collaborations among experts coming from the above indicated communities.

Interop-Vlab.IT is affiliated to *Interop-Vlab.Eu*, a network of research poles acting as an European virtual research laboratory in the interoperability domain. In Italy, the association has the goal to create synergies among different communities in order to reach new scientific results both at a national and international level on the themes of the systems interoperability. Furthermore, *Interop-Vlab.it* aims at ensuring technological transfer from research centers and universities to enterprises and public administration. The workshop aims at analysing and discussing the role of interoperability in the research community, its technological development and new interoperability solutions and applications. Then, current research and development activities, and running projects, are presented by the Interop-VLab.it partners.

The Steering Committee

Table of contents

Keynote Speech

Papers

Interoperability Framework: what's happening in Europe and in Italy

Francesco Tortorelli

DigitPA
tortorelli@digitpa.gov.it

Abstract

The presentation underlines the principles described in the most relevant and recent initiatives concerning interoperability at EU level. Moreover, it will be described the Interoperability Framework II and its finalization inside the European interoperability strategy. Finally, the presentation describes the Italian approach to the interoperability framework with solid legal and governance basis.

Collaborative enterprise knowledge mashup

Devis Bianchini, Valeria De Antonellis, Michele Melchiori

Università degli Studi di Brescia – Dip. di Ing. dell'Informazione
Via Branze 38 – 25123 Brescia (Italy)
{bianchin, deantone,melchior}@ing.unibs.it

Abstract. In this paper, we describe a proposal of semantic techniques to support enterprise mashup within or across collaborative partners. Mashups are Web applications that integrate data and/or application logics originated from third parties and made available through Web APIs. The aim of the presented techniques is to enable effective searching of mashup components and their composition, by making possible proactive suggestion of mashup components and progressive mashup composition. The approach, called SMASHAKER, includes a model of component semantic descriptor, techniques for building a component repository where semantic descriptors are semantically organized according to similarity and coupling links, and supports an exploratory perspective in mashup development.

1 Introduction

An enterprise mashup is defined as a Web-based application that combines existing content, data or services, from independent sources, by empowering also end users to create and adapt situational application to solve a specific problem. Enterprise mashup focuses on the User Interface integration by extending concepts of Service-Oriented Architecture with the Web 2.0 philosophy [3]. In mashup, data and services are made available through heterogeneous APIs. To better support developers during enterprise mashup development, it is crucial therefore abstract from underlying heterogeneity [1,4].

In this paper, we propose a novel conceptual approach to support progressive construction of collaborative enterprise mashups apt to combine multiple data and/or application logics. The approach is based on semantic annotation of components and semantic matching techniques for their organization, selection and composition.

2 Mashup construction in SMASHAKER

A mashup application is obtained by assembling, possibly with the minimum programming effort, available ready-to-use components. Generally speaking, mashup developing is a process composed of the following phases: (a) component selection from a repository or from the Web; (b) definition of event-operation associations and

I/O mappings among the selected components; (c) development of programming code to actually glue components and their user interfaces to get the final application. Our approach, called SMASHAKER, aims to supports the phases (a) and (b). The output of these phases is what we call a *conceptual mashup*, describing the selected components, associations and mapping. A recommendation system based on this development model should suggest to the developer the components that can be used as alternatives or that can be properly composed in the conceptual mashup.

Different roles must be considered in an enterprise mashup development context [3]:

- the *provider* of the mashup component, that is in charge of supplying the component description with its annotation to enable easy combination with other components;
- the *consumer*, who selects and combines the mashup components to build a conceptual mashup; we refer to this role in the following of the paper with a more specific term, *mashup designer*.

According to the SMASHAKER vision, the component APIs are semantically annotated, classified and made available to be assembled in a conceptual mashup through the following steps, schematically shown in Fig. 1.

Semantic annotation. Each available component is described by means of an annotation of its API. In this phase, the meanings of APIs are made explicit by associating API elements (inputs/outputs/operations) to concepts defined in domain ontologies. The result of this step is a collection of semantic descriptors.

Matching and linking of semantic descriptors. Semantic-based matching techniques are applied to the semantic descriptors previously defined to establish automatically similarity and coupling links between component descriptors.

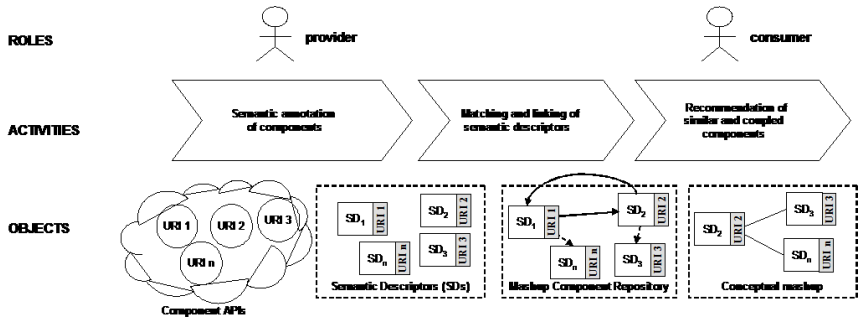

Fig. 1. The SMASHAKER approach to mashup development.

The links, as result of this phase, are stored in a *Mashup Component Repository (MCR)* to be available for the following step.

Component recommendation. Similarity and coupling links are exploited to obtain proactive recommendation of MCR components. In particular, in this step our framework enables: (i) proactive suggestion of component descriptors ranked with respect to their similarity with the mashup designer's requirements; (ii) interactive support to mashup designer for component composition, according to the exploratory perspective. The result of this step is a conceptual mashup, where component descriptors are properly connected.

3 The component semantic descriptor

To describe a component different elements must be considered. First, it must export a Web API, that is, a list of **operations** (methods signatures). For each operation, its I/O parameters are specified. Second, according to [1], integration of mashup components is typically event-driven: when the user interacts with the UI of components, it reacts with certain state changes and the other components must be aware of such changes to update their UIs accordingly. Each component has a set of **events** and event outputs. An event of a component can be connected to an operation of another component in a publish/subscribe-like mechanism. In a component *semantic descriptor (SD)*, names of operations, operation I/Os and event outputs are annotated with concepts from domain ontologies. Furthermore, a component is associated to a set of **categories**, to provide a domain-driven classification of the component itself.

As an example of component *semantic descriptor (SD)*, Fig. 2 shows a component called *MapViewer* for map visualization similar to the well known *Google Map*. The API of this component includes one operation to show a location on the map by specifying an address, city and country. Moreover, when the user clicks on the map to select a specific point, an event is triggered.

```
<SemanticComponent name="MapViewer_SD"
   url="http://www.mapview.com">

   <categories>
        <item>Mapping</item>
   </categories>
   <operation address="show"
      semanticReference="http://localhost:8080/Travel.owl#showLocation">
   <input
       semanticReference="http://localhost:8080/Travel.owl#Address"/>
   <input
       semanticReference="http://localhost:8080/Travel.owl#Country"/>
   <input semanticReference="http://localhost:8080/Travel.owl#City"/>
   </operation>
   ...
   <event address="selectedCoordinates">
   <output
       semanticReference="http://localhost:8080/Travel.owl#Coordinates"/>
   </event>
</SemanticComponent>
```

Fig. 2. An example of component semantic descriptor

4 The mashup component repository

In our approach, component semantic descriptors are organized in a Mashup Component Repository, to better support collaborative enterprise mashup. In the repository, descriptors are related in two ways: (i) semantic descriptors SD_i and SD_j of components which show an high relatedness between their I/O and therefore can be potentially wired in the final mashup application to the combine functionalities they offers, are connected through a *functional coupling link*; (ii) semantic descriptors SD_i and SD_j of components which perform the same or similar functionalities, are connected through a *functional similarity link*.

To identify coupling or similarity links (resp.), semantic matching techniques can be used. In particular, we have defined the *coupling degree coefficient* $Coupl_{IO}()$and the *functional similarity degree coefficient* $Sim_{IO}()$. These coefficients are based on the computation of name affinity $NA()$ between pairs of, respectively, (i) operations names, (ii) I/Os names and (iii) event outputs names used in the semantic descriptors to be matched [2]. $NA()$ evaluation is based both on a terminological (domain-independent) matching based on the use of WordNet and on a semantic (domain dependent) matching based on ontology knowledge.

In particular, $Sim_{IO}(SD_R, SD_C)$ between SD_R and SD_C is computed to quantify how much SD_C provides at least the operations and I/Os required in SD_R. and is maximum when SD_C provides at least the operations of SD_R.

$Coupl_{IO}(SD_i, SD_j)$ is maximum if every event ev in SD_i has a corresponding operation op in SD_j and, in particular, every output of ev has a corresponding input in op, no matter if SD_j provides additional operations.

4.1 Collaborative mashup developing

The MCR can be exploited for searching, finding and suggesting suitable components to be used in mashup developing. The designer starts by specifying a request SD_R for a component in terms of desired categories, operations and I/Os. A set of components SD_i which present a high similarity with the requested one and such that at least a category in SD_R is equivalent or subsumed by a category in SD_i are proposed. Components are ranked with respect to Sim_{IO} values. Once the consumer selects one of the proposed components, additional components are suggested, according to similarity and coupling criteria: (i) components that are similar to the selected one (the consumer can choose to substitute the initial components with the proposed ones); (ii) components that can be coupled with already selected ones during mashup composition. Each time the consumer changes and selects another component, the MCR is exploited to suggest the two sets of suitable components.

5 Conclusions

In this paper, we described a semantic framework for mashup component selection and suggestion for composition in the context of collaborative enterprise mashup.

Mashup components are semantically described and organized according to similarity and coupling criteria, and effective (semi-)automatic design techniques have been proposed.

6 References

1. Daniel, F., Casati, F., Benatallah, B. and Shan, M.C. (2009) Hosted Universal Composition: Models, Languages and Infrastructure in mashArt, *28th Int. Conference on Conceptual Modeling (ER09),* pages 428-443.
2. Bianchini, D., De Antonellis, V., Melchiori, M. (2008) Flexible Semantic-based Service Matchmaking and Discovery, *World Wide Web Journal*, 11(2):227-251.
3. Hoyer, V. and Stanoevska-Slabeva, K. (2009) Towards a Reference Model for grassroots Enterprise Mashup Environments, *17th European Conf. on Information Systems (ECIS)*.
4. Abiteboul, S., Greenshpan, O. and Milo, T. (2008) Modeling the Mashup space, *Workshops on Web Information and Data Management*, pages 87-94.

Clustering Enterprise Networks by Patent Analysis

Fulvio D´Antonio[1], Simone Orsini[1], Alessandro Cucchiarelli[1], Paola Velardi[2]

[1] Polytechnic University of Marche, Italy, email: cucchiarelli, dantonio, orsini@diiga.univpm.it
[2] Sapienza Università di Roma, Italy, email: dantonio,velardi@di.uniroma1.it

Abstract. The analysis of networks of enterprises can lead to some important insights concerning strategic aspects that can drive the decision making process of different players: business analysts, entrepreneurs, public administrators. In this paper we present the current development status of an integrated methodology to automatically extract enterprise networks from public textual data and analyzing them. We show an application to the enterprises operating in the Italian region of Marche.

Keywords: Social Network Analysis, Natural Language Processing, Clustering

1 Introduction

Networks of Enterprises [2] are a special kind of social networks in which the nodes represent enterprises and the links indicate some form of relationship among them.

The relationships that have been traditionally represented through links are business collaborations, enterprise similarity, mutual exchange of capitals, information flows, or hierarchical relationships like the ones representing supply chains or enterprises aggregation into districts.

Social Network Analysis [6] defines a number of measures and techniques that can be used for the evaluation and analysis of enterprise networks. Such measures, if examined by a business analyst, an entrepreneur or a public administrator can lead to some important insights concerning some strategic aspects of the network.

We describe here few scenarios in which the analysis can be conveniently applied:

- Domain analysis

The analyst inspects the network in order to understand which are the main productive sectors, the groups of similar enterprises, the relative strengths of such groups and their inter-relationships.

- Determining competitors

Mining non-cooperating similar enterprises which may be potential competitors in a given productive sector. There is either high or low level of competition? There is a potential for market penetration of my enterprise?

- Partnership discovery

Individuating similar or complementary enterprises aimed at establishing business/productive co-operations.

- Funds allocation

Analysis of productive trends and gaps, and setup of regional/national funding schemes.

But where the data about Networks of Enterprises come from?

The usual scenario is that the graph structure of the network is not explicitly available but has to be "distilled" from a dataset *D*, i.e., one has to infer the network structure starting from such data by applying some processing steps.

Let's examine, as an example, the case of networks whose (weighted) links represent the degree of "similarity" between the nodes. We have two possibilities:

1. We can submit questionnaires to the actors involved asking them to estimate their similarity with, let's say, one hundred of other enterprises. The similarity value could be a real number in the range [0,1], a set of symbols (sequence of stars, for example: * little, ** medium , *** high or no stars for no similarity) or similar representations.
2. If we have some textual data available, e.g. papers, websites, product manuals etc. we can use some form of natural language processing and information retrieval metrics to (semi)-automatically estimate the similarity.

The first approach is expensive, exposed to questionnaire's compiler subjectivity and implies a series of practical issues: distribution of the questionnaires, commitment to the questionnaire compilation in a given time and collection of the results.

The second approach enjoys the benefits of the general wealth of publicly available data and of automatic processing; everyone can search the web and obtain a great number of information (mainly textual) about the enterprises under examination. The drawbacks of this approach rely in the generally worse performance of natural language processing systems with respect to humans. Humans seems to be better in performing tasks like word-sense disambiguation, contextualizing judgement and understanding the textual information.

Hybrid approaches are also commonly adopted: an automatic NLP system interact from time to time with humans that take decisions about some harsh points.

Let's consider an enterprise interested in finding potential partners among the enterprises in a given geographical area, that, in turn, requires to find partners with similar interest. Even in small areas the enterprises, generally mostly SMEs, (Small-Medium Enterprises) can easily be in the order of several hundreds. If we decide to assign such task to a person we could apply the following strategy: we give him/her a list of some hundreds of enterprise names and some thousands of documents and related websites and we ask him/here to read the documents and surf the websites to extract key information about the business/productive sector of the enterprise in order to estimate from such information the degree of similarity and potential collaboration. This task is clearly not feasible for a human. A valid support can come from a carefully designed NLP system that can be supervised by the user and

occasionally corrected by him/her (e.g. eliminating non-relevant keywords in a particular domain, individuating uncaught spelling variation, etc).

2 Patent and Enterprise Networks

In this section we describe how we have distilled Networks of Enterprises starting from textual data publicly available about patents deposited by European enterprises.

The European Patent Office (EPO)[1] provides a uniform application procedure for individual inventors and companies seeking patent protection in up to 40 European countries. It is the executive arm of the European Patent Organisation and is supervised by the Administrative Council. Through its web-site and exposed web-services it is possible to access to information about European patents that have been registered; the information include, among the other things, the date of presentation, the applicant name and mission, the address of the applicant and the textual description of the patent.

The patents presented by an enterprise is a good indicator of the business sector in which the enterprise operates. Therefore through the EPO database we can gather textual data about the business/industrial sector of the enterprises in a given geographical location and we can use such data to extract similarity networks. The methodology we use is summarized in the following steps and it is similar to the ones used in [4,5]:

1. Gather patents registered by enterprises located in a given geographical area (a city, a region, a country, …);
2. Pre-process textual data to extract raw text;
3. Process raw text with a part-of-speech tagger;
4. Extract candidate annotating terms using a set of part-of-speech patterns [3];
5. Rank candidates, possibly filter them choosing a threshold [3];
6. Output a set of weighted vectors V of annotating terms for each documents;
7. Group the vectors by enterprise (that presented the patent applications) and construct a centroid (i.e. a mean vector) with such groups. This centroid roughly represents the business sector of the enterprise.
8. Build a graph computing a similarity function [1] for each pair of centroids.

2.1 Clustering

Data Clustering [8], originally conceived in the data mining field, is a very active research domain aiming at developing methods for dividing a set of data-points into subsets (called clusters) so that points in the same cluster are similar in some sense. We can use clustering techniques on our Enterprise Networks in order to discover potentially interesting networks patterns and to filter noisy phenomena.

[1] http://www.epo.org/

One of the main drawbacks of clustering is the substantial lack of possibility of validating results except for very special cases, e.g. when the distribution of data is known (like a multivariate Gaussian) or we have access to other forms of ground truth. In literature clustering validation is approached using internal and external validity criteria: the external criteria rely on comparison with available ground truth while the internal ones are constituted by metrics that estimate the internal coherence of a cluster (inter-cluster similarity) and its substantial dissimilarity from other clusters (intra-cluster dissimilarity). According to [7], each clustering technique should be evaluated in the context of a micro-economic setting, i.e. in maximizing an objective function.

We relax as much as possible the notion of clustering: given a set A, a clustering C is a set of subsets of A, i.e. $C \subseteq \mathrm{P}(A)$ where P(A) is the power set of A. A *crisp clustering* is a clustering with pairwise disjoint clusters and a *partitive clustering* is when the union of clusters is A ($\bigcup_{C_i \in C} C_i = A$).

Most of the clustering techniques developed concentrate on producing *partitive crisp clusterings*.

2.2 Graph clustering by mean of components density maximization

In this paper we use a very simple algorithm for graph clustering. Given a graph $G=(V,E)$ in which V is a set of vertices and E is a set of weighted edges (x,y,w) with x,y in V e w in $[0,1]$, we order the edges in E with respect to the weights obtaining the sequence $e_1,\ldots,e_{|E|}$. We then construct the sequence of graphs $GS=G_0,\ldots G_{|E|}$ in which $G_i=(V,\{e_1,\ldots,e_i\},$ i.e. the i-eth graph is the graph containing the top-i weighted edges. The clusters are the connected components of each graph and each graph contains all the others following in the sequence so that, therefore, we have a hierarchical clustering.

To choose a representative of this sequence we maximize the function scoring the *mean components density*: for a graph we compute the density of each connected component, we sum them and we divide by the number of components. The (weighted) density of a connected graph is:

$$d(G) = \frac{\sum_{(x,y,w)\in E_G} w}{\binom{|V|}{2}}$$

The *mean components density* is:

$$meand(G) = \frac{\sum_{C \in Components(G)} d(C)}{|Components(G)|}$$

And finally, we can choose the preferred clustering G_{pref} by maximizing *meand*:

$$G_{pref} = \underset{Gi \in GS}{\arg\max}\, meand(G_i).$$

3 Applications

In figure we show a detail of the graph obtained by applying the described method to the enterprises operating in the Italian region of Marche that registered European patents. The graph has been clustered according to the algorithm in section 2.2.

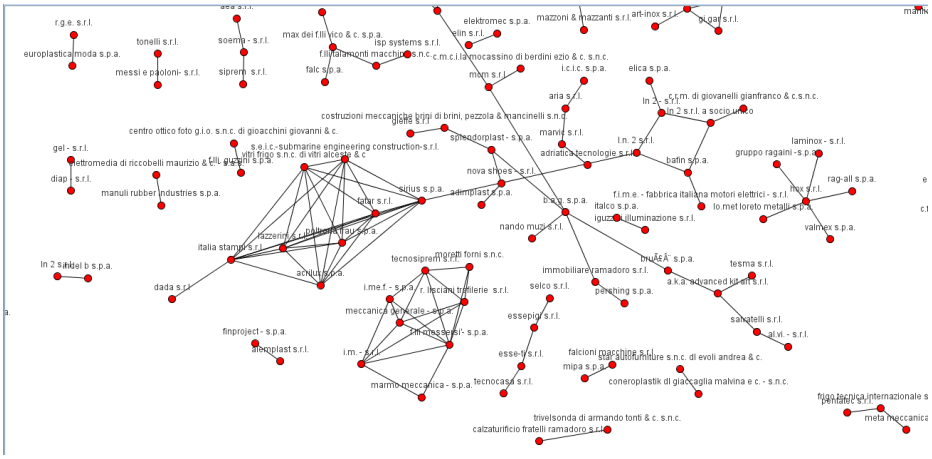

Fig. 1. The Network Of Enterprises of Region Marche (detail)

In the figure we can visually locate a very dense cluster in the middle-left; unfortunately an in deep analysis of this clusters reveals that it is consisting of all enterprises that deposited patents in German language. At the beginning of the experimentation we didn't notice that some patents descriptions are not written in English language. This noisy phenomenon, anyway, emerged because of clustering and we suggest that this can become one important use of clustering techniques: locating "spam" clusters in order to eliminate them and iteratively refine the process.

In the rest of the picture we notice a high degree of fragmentation: several very small groups (2 or 3 elements) and rare bigger groups.

We report here some examples of clusters:

- Moretti forni S.p.a
- Defendi Italy S.r.l
- Officine Meccaniche Defendi S.r.l
- S.o.m.i press

In which the similarity links depend mainly on the terms: *gas, flame, burner, cooking*. We can suppose this is a cluster consisting of cooking-furniture enterprises.

Another cluster is constituted by:

- Best S.p.a
- Gitronica S.r.l
- Intec-s.r.l

depending on the terms *phone, microphone, voice, electronic component*.

In general is very difficult to evaluate the quality of the produced clusters and we performed only a qualitative analysis.

A high level of fragmentation is, indeed, a problem. The utility of clustering in general is to reduce the dimension of problems: if the number of clusters is comparable with the number of elements we haven't performed any reduction at all and the clustering is useless. As we performed just an initial experimentation we are not able to say if the fragmentation observed is a real phenomenon in the application domain or can be reduced by refining the techniques used in the various steps of the process.

Therefore, in the future, we plan to work on the following points:

- The NLP analysis tools and techniques we adopt are powerful enough to put in light important similarities/differences in the domain studied?
- The data used are enough complete/noise-free/etc? If not, how can we perform data cleaning and gather additional data?
- The clustering method proposed is comparable with respect to state-of-the-art methods?

4 References

1. Ricardo Baeza-Yates and Berthier Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1st edn., May 1999.

2. T. Elfring and W. Hulsink, 'Networking by entrepreneurs: Patterns of tie-formation in emerging organizations', Organization Studies, 28(12), 1849–1872, (2007).

3. Francesco Sclano and Paola Velardi, 'Termextractor: a web application to learn the shared terminology of emergent web communities', in Proceedings of the 3rd International Conference on Interoperability for Enterprise Software and Applications (I-ESA 2007), Funchal, Portugal, (2007).

4. Paola Velardi, Alessandro Cucchiarelli, and Fulvio D'Antonio, 'Monitoring the status of a research community through a knowledge map', Web Intelli. and Agent Sys., 6(3), 273–294, (2008).

5. Paola Velardi, Roberto Navigli, Alessandro Cucchiarelli, and Fulvio D'Antonio, 'A new content-based model for social network analysis', in ICSC '08: Proceedings of the 2008 IEEE International Conference on Semantic Computing, pp. 18–25, Washington, DC, USA, (2008). IEEE Computer Society.

6. Stanley Wasserman, Katherine Faust, and Dawn Iacobucci, Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences), Cambridge University Press, November 1994.
7. J. Kleinberg, C. Papadimitriou, P. Raghavan. A micro-economic view of data mining. Data Mining and Knowledge Discovery, 2(4), 1998.
8. Ian H. Witten, Eibe Frank , Data Mining: Practical Machine Learning Tools and Techniques (Second Edition), Morgan Kaufmann June 2005

Reasoning on Business Processes and Ontologies in a Logic Programming Environment

Michele Missikoff[1], Maurizio Proietti[1], Fabrizio Smith[1,2]

[1] IASI-CNR, Viale Manzoni 30, 00185, Rome, Italy
[2] DIEI, Università degli Studi de L'Aquila, Italy
{missikoff, maurizio.proietti, fabrizio.smith}@iasi.cnr.it

Abstract. In this paper we present a semantic approach to Business Process (BP) Management. The proposal is based on a synergic use of an ontological framework (OPAL), to capture the semantics of a business scenario, and a business process modelling framework (BPAL), to represent the underlying application logic. Both frameworks are grounded in a logic-based formalism (Logic Programming) and therefore it is possible to apply effective reasoning methods to make inferences over a BPKB (Business Process Knowledge Base) stemming from the fusion of the two.

1 Introduction

Business Process (BP) management is constantly gaining popularity in various industrial sectors and in the public administration. But, despite the growing academic interest and the penetration in the business domain, heterogeneous and ad-hoc solutions that often lack a formal semantics have been so far proposed to deal with several arisen issues, such as: cross-enterprise integration and collaboration, adoption of organizational and data models in conjunction with workflow models, query and retrieval of BP fragments, BP composition.

In order to increase the level of automation in the specification, analysis, implementation and monitoring of BPs, various papers have advocated the enhancement of BP management tools by means of well-established techniques from the area of the Semantic Web, like, for instance, computational ontologies [1,2]. The use of an ontology allows an unambiguous definition of the entities occurring in the domain, and eases the interoperability between software applications and the reuse/exchange of knowledge between human actors. However, there are still several open issues regarding the combination of workflow languages (with their specific execution semantics) and ontologies, and the accomplishment of reasoning tasks involving both these components.

In this paper we present a logic-based framework that aims at providing a uniform and formal representation of both the behavioral (i.e., workflow-related) and the structural (i.e., ontology-related) domain knowledge about a business process. Our framework is also equipped with a powerful inference mechanism supported by the tools developed in the area of Logic Programming [3].

1 Business Process Knowledge Representation

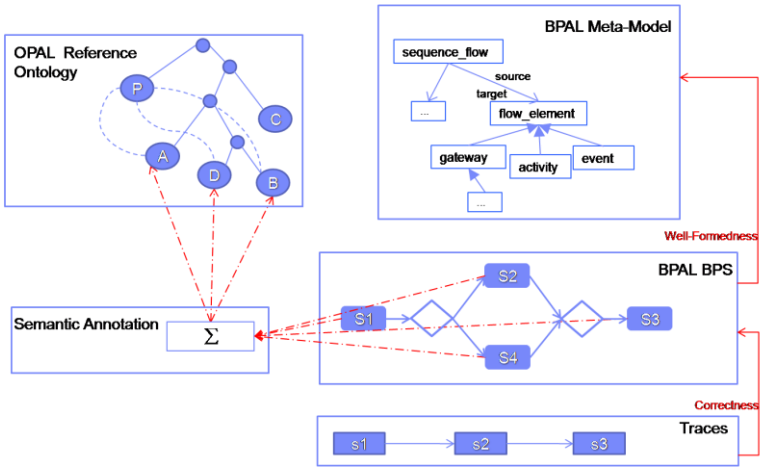

Fig. 1. Business Process Knowledge Base

The knowledge about business processes and the context where they operate, is stored in a *Business Process Knowledge Base (BPKB),* as exemplified in Figure 1 and briefly described in the following.

OPAL [4] is an ontology representation framework supporting business experts in building a structural ontology, where concepts are defined in terms of their information structure and static relationships. OPAL provides a set of upper level concepts and a set of design principles (patterns) to capture the active entities (actors), passive entities (objects), and transformations (processes). A significant core of an OPAL ontology can be formalized by a fragment of OWL, by using the OWL-RL [5] profile. OWL-RL, is an OWL subset designed for practical implementations using rule-based technologies such as logic programming [6].

BPAL [7] (Business Process Abstract modelling Language), is a logic-based language (grounded in Horn Logic) that provides a declarative modeling method capable of fully capturing the procedural knowledge in a business process. BPAL constructs are based on the BPMN 2.0 specification [8] and provide a comprehensive modelling method that spans from the ground level (to model the traces that are produced by the execution of a BP), to the BP schema (BPS) modelling level (where the designer actually defines the diagram that represents the business logic of the BP), to the meta-modelling level (the basic formation rules that guide the designer in the specification of the BP schema).

Semantic Annotation is a correspondence between elements of the BP schema and elements of the Reference Ontology specified using the 'sigma' predicate. It consists of a set of assertions of the form $\sigma(Act,Conc)$, where *Act* is a constant that denotes an entity of a BP schema, and *Conc* is a constant used to denote a concept defined in the ontology. This relation allows us to specify the meaning of the entities of a business process in terms of a suitable conceptualization of the domain of interest.

2 Reasoning with the Business Process Knowledge Base

The components of the *BPKB* introduced in the previous section are formalized by a First Order Logic theory, defined as

$$\boldsymbol{BPKB} = \boldsymbol{BRO} \cup \boldsymbol{\Sigma} \cup \boldsymbol{M} \cup \boldsymbol{B} \cup \boldsymbol{T}$$

where: ***BRO*** is an OPAL Business Reference Ontology; **Σ** is the semantic annotation, i.e. a set of assertions of the form $\sigma(Act,Conc)$; ***M*** is the theory formalizing the meta-model and the related notion of well-formedness of a BP schema; ***B*** is a set of BPAL BP schemas, i.e. a set of assertions (ground facts) constructed from the BPAL alphabet; ***T*** is the theory formalizing the trace semantics of a BP schema and the notion of correctness of a trace w.r.t. that schema.

A relevant property of the *BPKB* is that it has a straightforward translation to a logic program [3], which can be effectively used for reasoning within a Prolog environment. This translation allows us to deal within a uniform framework with several kinds of reasoning tasks and combinations thereof, within the uniform framework of logic programming. Every component of the *BPKB* defines a set of predicates that can be used for querying the knowledge base. The reference ontology ***BRO*** and the semantic annotation **Σ** allow us to express queries in terms of the ontology vocabulary. The predicates defined by the meta-model theory ***M*** and by the BP schemas ***B*** allow us to query the schema level of a BP, verifying properties regarding the flow elements occurring in it (*activities, events, gateways*) and their relationships (*sequence flows*). Finally, the predicates defined by the trace theory ***T,*** allow us to express queries about the behavior of a BP schema at execution time, i.e., verify properties regarding the execution semantics of a BP schema.

In order to provide the user with a simple and expressive query language that does not require to understand the technicalities of the logic engine, we proposed in [9] *QuBPAL*, a simple query language based on the SELECT-FROM-WHERE paradigm that can be translated to Prolog[2] queries for their evaluation. As example we report in the following a *QuBPAL* query:

*SELECT <?p,?s,?e >FROM **
WHERE activity(?s::ReceivingPO), activity(?e::Delivering),
precedence(WaitingClearence,Delivering,?p,?s,?e)

This query returns all the well-formed process fragments (i.e., structured blocks [7]) such: (i) start with an activity of *ReceivingPO* (i.e., an activity annotated with the concept *ReceivingPO*), (ii) end with an activity of *Delivering,* and (iii) contain an activity of *WaitingClearence* which is always executed (not necessary immediately) before *Delivering.* The SELECT statement defines the output of the query evaluation, which in this case is a process fragment identified by the triple *<?p,?s,?e>*, where *?p* is a BP identifier, *?s* is the starting element, and *?e* is the ending element. The FROM statement indicates the process(es) from which data is to be retrieved, in this case "*"

[2] In particular queries not involving ***T*** are translated to Datalog queries with stratified negation.

stands for the whole repository. In the **WHERE** statement it can be specified an expression which restricts the data returned by the query.

3 Implementation

A prototype of the proposed framework has been implemented as a Java application, interfaced with the XSB logic programming and deductive database system [10]. The BPAL platform is depicted in Figure 2. On the left part of this figure, enclosed in a dotted line, we have are grouped the components involved in the *set up phase*, where the BPKB is built.

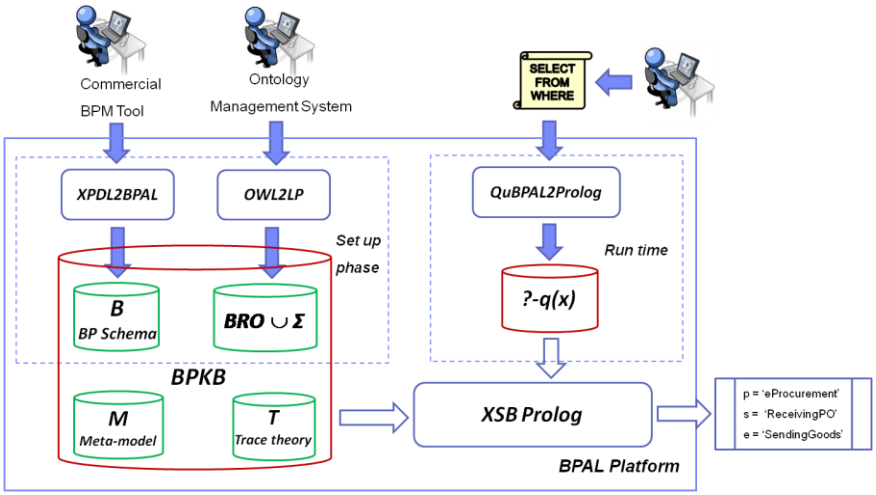

Fig. 2. Architecture of the BPAL Platform

The process repository ***B*** is populated by process schemas modeled by business experts in XPDL (XML Process Definition Language [11]) and translated into BPAL by means of the service *XPDL2BPAL*. The business reference ontology ***BRO*** is imported from the Athos OPAL Ontology Management System [12] and is added to the *BPKB* together with the semantic annotation **Σ** of the BP schemas**.** Both ***BRO*** and **Σ** are represented in the OWL language. OWL ontologies (restricted to the RL profile) are imported into the *BPKB* in the triple notation by the service *OWL2LP*. A Prolog translation of the OWL 2 RL/RDF rules [5] is also included in the *BPKB* to implement reasoning over the ontology. The *BPKB* is completed by the logic programs encoding the meta-model theory ***M***, and the trace theory ***T***. After the population of the *BPKB* the reasoning tasks can be performed at *run time* by querying the knowledge base through *QuBPAL* queries, that are translated into Prolog by the service *QBPAL2Prolog* and evaluated as goals by the XSB engine. These component are enclosed in a dotted rectangle on the right part of Figure 2.

4 Conclusions

In this paper we presented the main ideas of a framework conceived to complement existing business modeling tools by providing advanced reasoning services. The proposed platform consists of several parts: (i) an ontological framework, OPAL, to capture the semantics of the business scenario; (ii) a business process modeling framework, BPAL, to capture the application logic; (ii) a reasoning engine, based on Logic Programming, that operates on the above two structures in an integrated way; (iv) a BP query language, developed on top of the reasoning engine; finally, (v) a verification mechanism, tightly connected to the latter.

The discussed *BP Knowledge Base* constitutes the base of a knowledge representation framework that we want to extend in several directions. First of all by handling any graph-structured BP schema (without the blocked assumption), and hence the verification of behavioral properties over (possibly) infinite sets of traces. We are also investigating the verification at *run time* (i.e. over a running instance of the process during its enactment) and the *a-posteriori* analysis (i.e., log mining) over the information stored during the execution. On an engineering ground, we are exploring the problem of manipulating, merging and aggregating a set of business process fragments in the contexts of BP Composition and BP Re-engineering.

5 References

1. Hepp, M., Leymann, F., Domingue, J., Wahler, A., Fensel., D.: Semantic business process management: A vision towards using semantic web services for business process management. In: ICEBE 2005, pp. 535–540. IEEE Computer Society, Los Alamitos (2005).
2. Roman, D., et al.: Web Service Modeling Ontology. Applied Ontology, 1(1): 77-106, 2005.
3. Lloyd, J.W.: Foundations of Logic Programming. Springer-Verlag, Berlin, 1987.
4. D'Antonio, F., Missikoff, M., Taglino, F.: Formalizing the OPAL eBusiness ontology design patterns with OWL. In the 3rd I-ESA Conference, 2007.
5. OWL 2: Profiles, http://www.w3.org/TR/owl2-profiles.
6. Grosof, B. N., Horrocks, I., Volz, R., Decker, S.: Description Logic Programs: Combining Logic Programs with Description Logic. In the Proc. of WWW 2003, Budapest, Hungary.
7. De Nicola A., Missikoff M., Proietti M., Smith F.: An Open Platform for Business Process Modeling and Verification, International Conference on Database and Expert Systems Applications, DEXA, September 2010. LNCS 6261 Springer 2010.
8. Business Process Model and Notation V. 2.0, 2009, http://www.omg.org/spec/BPMN/ 2.0.
9. Missikoff M., Proietti M., Smith F. A Business Process Knowledge Base for Composite Services Development. International Workshop BSME, Malaga, June 2010.
10. The XSB Logic Programming System. Version 3.1, Aug. 2007, http://xsb.sourceforge.net.
11. XPDL 2.1 Complete Specification, Oct. 2008, http://www.wfmc.org/xpdl.html.
12. ATHENA, D.A3.2 "Updated version of the Ontology Authoring and Management System with semantic search functions", ATHENA IP, deliverable, 2005.

A Semantic Clouding Approach for Cross-Webs Interoperability

Silvana Castano, Alfio Ferrara, Stefano Montanelli, Gaia Varese

Università degli Studi di Milano
Dipartimento di Informatica e Comunicazione
Via Comelico, 39 – 20135 Milano, Italy
{castano, ferrara, montanelli, varese}@dico.unimi.it

Abstract. The classical vision of the Web as a merely publishing environment for information-consuming users is being replaced by a plural vision where multiple webs, like Web 2.0, Social Web, and Semantic Web, co-exist and interoperate to make information sharing more effective and socially pervasive. In this paper, we propose a *semantic clouding approach* for the construction of cross-web, disciplined, and intuitive information organization structures called *i*-clouds. An overview of the proposed semantic clouding approach is presented in the paper, as well as an example of *i*-cloud over real web resources about a movie dataset.

1 Introduction

Over the recent years, the classical vision of the Web as a merely publishing environment for information-consuming users is being replaced by a plural vision where multiple webs, like Web 2.0, Social Web, and Semantic Web, co-exist and interoperate to make information sharing more effective and socially pervasive. The experience of active research projects like OKKAM and Linked Data places the accent on the growing need to recognize identity and similarity relations between data descriptions provided by different web sources in different domains. The variety of webs data, spanning from textual tags to RDF(S) structural descriptions up to formal OWL instances, makes the above mentioned need of identity/similarity recognition even more crucial and challenging. In such a complex scenario, a new generation of information search techniques is required to cope with the following needs: i) the capability to span across multiple Webs, to properly consider the wide variety of available web resources and pieces of knowledge by properly assessing their information contribution nature; ii) the capability to anticipate the user needs by providing a focused but comprehensive set of web resources prominent for his/her target; iii) the capability to semantically organize all retrieved prominent resources into an intuitive and coherent structure [1,2].

In this paper, we propose a *semantic clouding approach* for the construction of cross-web, disciplined, and intuitive information organization structures called *i*-

clouds. An *i*-cloud is built to organize all the web resources about a certain *target entity of interest* into a graph on the basis of their level of *prominence* and reciprocal *closeness*. An overview of the proposed semantic clouding approach is presented in the paper, as well as an example of *i*-cloud over real web resources about movies. A more technical discussion about construction and formal properties of *i*-clouds is provided in [3].

2 Semantic clouding of web resources

An *i*-cloud is built around a certain target entity, which is a keyword-based representation of a topic of interest, namely a real-world object/person, an event, a situation, or any similar subject that can be of interest for the user. The notions of *closeness*, and *prominence* are define for an *i*-cloud to capture how similar web resources are each other and with respect to the target entity of the *i*-cloud, and the relative importance of a resource within the *i*-cloud, respectively. The following properties characterize *i*-clouds:

- *Cross-webness*. An *i*-cloud collects web resources coming from multiple webs to provide a comprehensive picture of all the available information, both objective and subjective, about the specified target entity for which the *i*-cloud is built.
- *Discipliness*. The web resources in an *i*-cloud are not only those directly related to target entity (i.e., those trivially matching the target entity) but also those that are in some way related to the target and are close to it.
- *Intuitiveness*. The *i*-cloud organization borrows the graphical representation commonly used for folksonomies and tag-clouds. This supports the user in browsing the *i*-cloud more effectively according to closeness and prominence of the web resources therein contained.

For *i*-cloud construction, we propose a semantic clouding approach articulated in three main phases (see Fig. 1): *acquisition of web resources*, *classification of web resources*, and *clouding of web resources*.

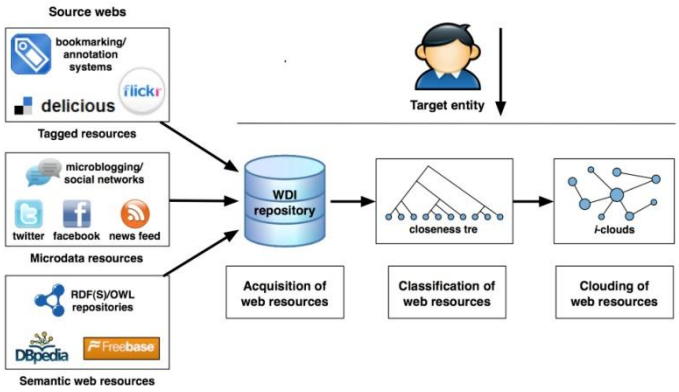

Fig. 2. The semantic clouding approach

Acquisition of web resources. For semantic clouding, all the different web resources are acquired from their respective source webs according to a reference data model called *WDI model*. The WDI model is capable of dealing with a variety of web resources. In particular, a WDI representation is provided for *tagged resources* that are resources from social annotation systems like Delicious, *microdata resources* that are resources from microblogging systems like Twitter, and *semantic web resources* that are resources from RDF/OWL knowledge repositories like Freebase. Each web resource *wr* is stored in a support repository called WDI repository in the form of a *web data item wdi(wr)* where terminological, structural and logical information about *wr* are properly represented.

Classification of web resources. The acquired web resources are grouped together according to their level of closeness. To this end, tailored matching techniques have been developed in the framework of the HMatch 2.0 system [4]. First, term and structural matching techniques are adopted to calculate the level of closeness $CC(wdi_i, wdi_j)$ between any pair of web data items wdi_i and wdi_j in the WDI repository. Then, a hierarchical clustering procedure is adopted to determine a *closeness tree* where all the wdis are properly grouped according to their closeness coefficients previously calculated.

Clouding of web resources. Given a target entity *e* specified by the user, an *i*-cloud is built for *e* by firstly extracting from the WDI repository a *ground set* of web data items that syntactically match *e*. Given a closeness threshold, the wdis in the ground set are used to select a number of candidate clusters in the closeness tree, namely all the clusters containing the wdis of the ground set and the wdis whose closeness is greater than or equal to the threshold. The candidate clusters originate the graph structure of the resulting *i*-cloud through a graph construction procedure. A labelling function is finally applied to assign to nodes and edges of the *i*-cloud their corresponding closeness and prominence values, respectively.

3 An example of *i*-cloud for cross-web interoperability

As an example, we consider the *i*-cloud of Fig. 2 where a number of web resources related to the target entity "Star Wars" are shown. We can observe that resources in the *i*-cloud are not only those directly related to this popular movie, such as the titles of the six movies of the Star Wars saga, but also resources that are close to the movie saga even if not directly matching the target, such as some of the most important characters in the movies. The dimension of each node in the *i*-cloud is proportional to the prominence of the corresponding web resource for "Star Wars" and the edges connecting the nodes are labelled with their closeness degree. We observe that different kinds of web resources populate the "Star Wars" *i*-cloud. In particular, this *i*-cloud is built over resources acquired from the Delicious annotation systems (e.g., wdi(delicious1)), the Twitter microblogging system (e.g., wdi(twitter1)), and the Freebase Linked Data repository (e.g., wdi(iimb1)).

In this example of *i*-cloud, the prominence of the various web resources is calculated through a *popularity-based* mechanism. This means that the prominence of a resource *wr* depends on the "centrality" of *wr* with respect to the *i*-cloud and it corresponds to the degree of connection of *wr* with the other nodes in the graph of the *i*-cloud [5]. Other techniques can be used for prominence computation based on the provenance of the web resources in the *i*-cloud (e.g., [6]).

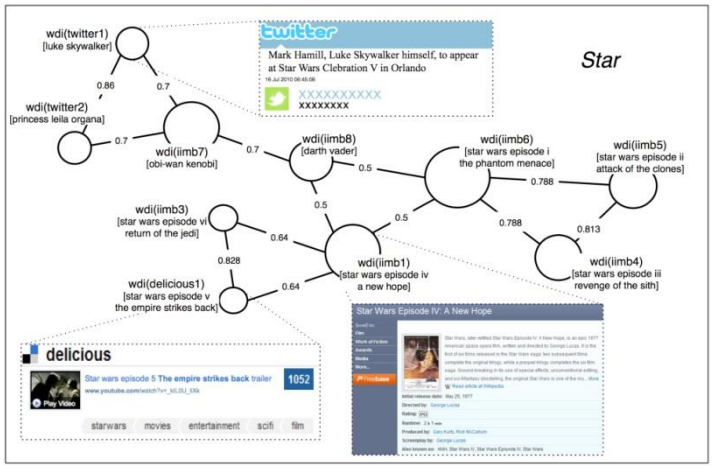

Fig. 3. Example of *i*-cloud for the entity "star wars".

4 Concluding remarks

A more detailed description of the approach and related support techniques can be found in [3]. A prototype for *i*-cloud construction has been developed on top of the HMatch 2.0 environment (http://islab.dico.unimi.it/hmatch) and it has been evaluated on the OAEI-IIMB 2010 dataset (http://www.instancematching.org/oaei/imei2010.html). The

positive results we obtained during evaluation encourages to continue working on *i*-cloud research issues. In particular, a focused search application is being developed in the domain of tourism and entertainment related to the city of Milan.

5 References

1. Kuo, B., Hentrich, T., Good, B., Wilkinson, M.: Tag Clouds for Summarizing Web Search Results. In: *Proc. of the 16th Int. Conference on World Wide Web (WWW 2007)*. Banff, Alberta, Canada, 2007.
2. Koutrika, G., Zadeh, Z., Garcia-Molina, H.: Data Clouds: Summarizing Keyword Search Results over Structured Data. In: *Proc. of the 12th Int. Conference on Extending Database Technology (EDBT 2009)*. Saint Petersburg, Russia, 2009.
3. Castano, S., Ferrara, A., Montanelli, S.: Semantic Data Clouding across Multiple Webs. Submitted to *Information Systems*. Elsevier, 2010.
4. Castano, S., Ferrara, A., Montanelli, S.: Dealing with Matching Variability of Semantic Web Data Using Contexts. In: *Proc. of the 22nd Int. Conference on Advanced Information Systems Engineering (CAiSE'10)*. Hammamet, Tunisia, 2010.
5. Easley, D., Kleinberg, J.: Networks, Crowds, and Markets: Reasoning About a Highly Connected World. Cambridge University Press, 2007.
6. Gil, Y., Artz, D.: Towards Content Trust of Web Resources. *Web Semantics: Science, Services and Agents on the World Wide Web* 5(4), 2007.

Hierarchical Clustering of Process Schemas

Claudia Diamantini, Domenico Potena

Dipartimento di Ingegneria Informatica, Gestionale e dell'Automazione "M. Panti",
Università Politecnica delle Marche - via Brecce Bianche, 60131 Ancona, Italy
{diamantini,potena}@diiga.univpm.it

Abstract. In this work, we focus on the analysis of process schemas in order to extract common substructures. In particular, we represent processes as graphs, and we apply a graph-based hierarchical clustering technique to group similar sub-processes together at different levels of abstraction. We discuss different representation choices of process schemas that lead to different outcomes.

1 Introduction

Process Mining (PM) is the application of inductive techniques to extract general knowledge about business processes from process instances. In state of the art research, instances are traces of running processes recorded in the event logs of ERP, Workflow Management Systems or other enterprise systems, and the goal of PM is to distill a structured process description, from the set of real executions, representing the *process schema* [3]. This mining activity can be exploited for instance to support process mapping activities. In this paper we consider a different process mining task: given a set of process schemas, find groups of similar (sub-) processes. In order to achieve this task, we discuss the application of SUBDUE [1], a hierarchical graph clustering algorithm. Graph clustering techniques have been considered since process schemas have a inherent graph structure, while hierarchical clustering in general, and SUBDUE in particular, allows to account for the inherent abstraction structure typical of processes (from very general macro-processes down to simple activities). Although process schemas can be seen as graphs, the application of SUBDUE requires some choices in terms of how to represent complex flow control structures, like parallel and alternative execution of activities or merging. Sections 2 and 3 discuss different representation choices and their experimental evaluation. Section 4 briefly discusses the results and possible applicative scenarios.

2 Methodology

Given a set of directed graphs $G_i = <N_i, A_i>$ where N_i is the set of nodes and $A_i \subseteq N_i x N_i$ is the set of (possibly labeled) arcs, SUBDUE generates a clustering lattice of typical substructures. In its exact matching version, graphs are iteratively analyzed to discover at each step a cluster of isomorphic substructures. The cluster is then used to

compress the graphs, by substituting to each occurrence of the substructure a single node. The compressed graphs are presented to SUBDUE again, and the process is repeated until no more compression is possible. The output clusters turn out to define a lattice where the clusters are linked if a cluster appears in the definition of another. At each step, the substructure is chosen on the basis of its compression capability, measured by the Minimum Description Length (MDL) heuristics. The description length of a graph is measured by the number of bits needed to represent its adjacency matrix. The algorithm has been successfully applied to analyze structured objects in several domains (see http://ailab.wsu.edu/subdue/) thanks to the flexibility it gives to represent complex objects in terms of mathematical graph structures, and suggesting it as a promising technique to analyze process schemas.

A process schema describes the flow of work performed by a certain number of actors. The kinds of flow include simple sequences of activities (SEQ), and operators used to model parallelization (hereafter called SPLIT) and merging (JOIN) of activities. In particular, a SPLIT-AND means that the end of an activity starts all the linked activities, while in a SPLIT-XOR only one will be executed. Symmetrically, a JOIN-AND indicates that an activity begins when all the previous activities are terminated, while in a JOIN-XOR the completion of a single activity is needed. Figure 1 shows an example of process using some of the described operators in BPMN notation.

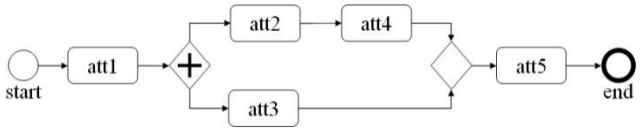

Fig. 4. An example of process schema. Activity *att1* is followed by both *att2* and *att3* (SPLIT-AND), and *att5* is started when *att4* or *att3* are completed (JOIN-XOR).

The application of SUBDUE to business processes requires to perform a mapping from the richer process graph to simpler directed graphs. As we will see, different representation choices may influence the final clustering result. While it seems straightforward to represent the SEQ operator by an arc in the graph, the representation of other operators is not straightforward. We present here three different models, named A, B, and C respectively, and characterized by an increasing level of compactness of the graph, without loss of information. In the A model, any operator is represented by a node called *operator*, which is linked to another node specifying the AND or XOR nature of the operator. In this model join and split are distinguishable by the number of ingoing and outgoing arcs (one outgoing arc and several ingoing arcs for join, the opposite for split). In the B model the node *operator* is replaced by different nodes one for each kind of operator. Finally, the C model simplifies the graph by removing both join and split nodes: since JOIN-XOR and SPLIT-XOR operators represent different alternative executable paths, one for each ingoing (outgoing) activity of a join (split) operator, XOR nodes can be removed by individuating all the possible alternative paths in the process, and generating a graph for each path. In this way, there is no ambiguity about the AND nature of arcs leaving

(entering) a node, so AND nodes can be removed too. Figure 2 shows the representation of the process in Figure 1 with respect to A, B and C models. Note that the three representations hold the same information, and the last produces two compact graphs (one for each xor path). Note also the use of labeled arcs in the C model of Figure 2 to maintain information about domain and range nodes. This is necessary to guarantee the correct interpretation of the final lattice after the compression performed by SUBDUE. It is straightforward to see that these representation strategies can be simply extended to include other BPMN constructs as well (in fact, the first two are directly related to the approach presented in [2]).

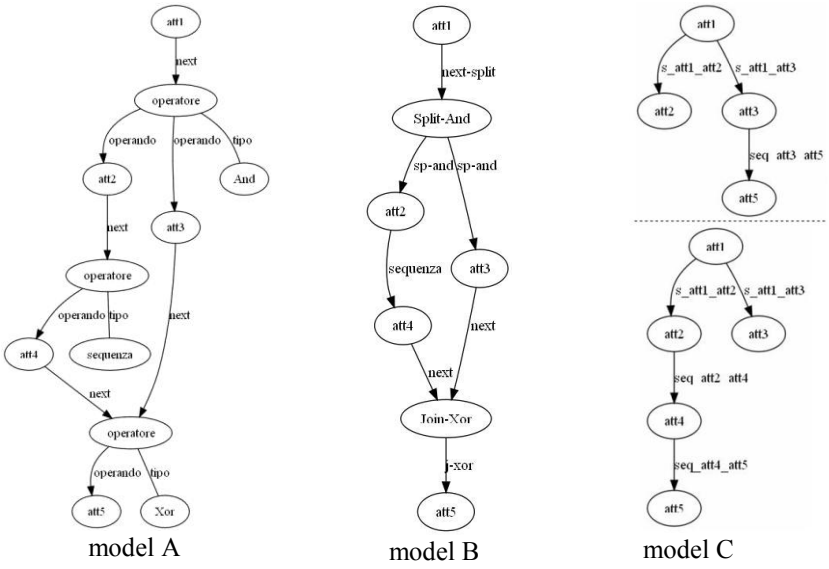

Fig. 5. The representation of the process schema in Figure 1 in conformity with the three proposed models

3 Experimental Evaluation

We experimented the methodology on a set of prototype processes describing e-science activities. In particular, we use a set of data mining processes for the classification task produced in the KDDVM project (http://boole.diiga.univpm.it). Activities are chosen among 21 algorithms of different kind (classification, pre-processing and post-processing) to generate a set of 40 different prototype processes.

In order to evaluate the resulting SUBDUE lattice with different representation strategies and the potentiality of the approach, we introduce some indexes: completeness, representativeness and significance. *Completeness* measures the

number of original graph elements still present in the final lattice[3]. It is expressed as $C = \frac{N_O + A_O}{N_I + A_I}$, where I is the set of input graphs and O is the final lattice. Node completeness is also considered. While completeness measures a quality of the whole lattice, the other indexes allows to individually evaluate each cluster. The *representativeness* of a substructure measures the number of input graphs holding the given substructure at least once. More precisely, representativeness of the substructure S_i is: $R(S_i) = \frac{G(S_i)}{G}$, where $G(S_i)$ is the number of processes holding S_i in graph G. High values of $R(S_i)$ indicate S_i as a typical subprocess. Finally, *significance* is a qualitative index that evaluates the meaning of a cluster with respect to the domain. This index allows us to disregard those clusters that are very representative, but do not contain useful knowledge. In Table 1, we synthetically show results of experimentations in terms of indexes values. In particular, clusters indexes are reported only for high level clusters, which represent the most common substructures. From Table 1, it results that all models are characterized by high completeness, even if C model leads to a slight decrease in the value of such index. The low significance of top level clusters obtained using A model is due to the fact that most frequent substructures are nodes representing individual operators, without references to involved activities. The highest values of representativeness for A model also depends on the high frequency of top level clusters. The C model is that allowing to achieve overall best results, reporting as top level clusters high-frequency substructures that are common in input graphs and are significant in the domain: they are actually knowledge patterns.

Figure 3 shows some of these knowledge patterns. We can see that the most used classification algorithms in the set of data mining processes are BVQ and C4.5. Furthermore, the practice of applying pre-processing algorithms to remove missing values and reduce the dimensionality of datasets emerges as typical patterns. We conclude by noting that SUB_9 and SUB_4 enlighten a not well-formed pattern, since removeMissingValue is performed after LDA. This is not a clustering error, rather it enlighten some problems in input process schemas.

[3] As a matter of fact, during the lattice generation, SUBDUE discards those substructures having low compression capability. This may lead to loose some node or arc.

	A Model	B Model	C Model
Completeness	97%	94%	92%
Nodes Completeness	99%	99%	98%
Representativeness of high level clusters	7%– 67%	8%-31%	8%-40%
Significance of top level clusters	- -	+	+ +

Table 1. Comparison of lattices obtained from graphs represented in accordance with the A, B and C models.

4 Discussion

The paper presents preliminary results about the feasibility of a graph-based clustering approach to recognize similarities among business processes, and to select significant prototypes. In particular, different representation alternatives of a business

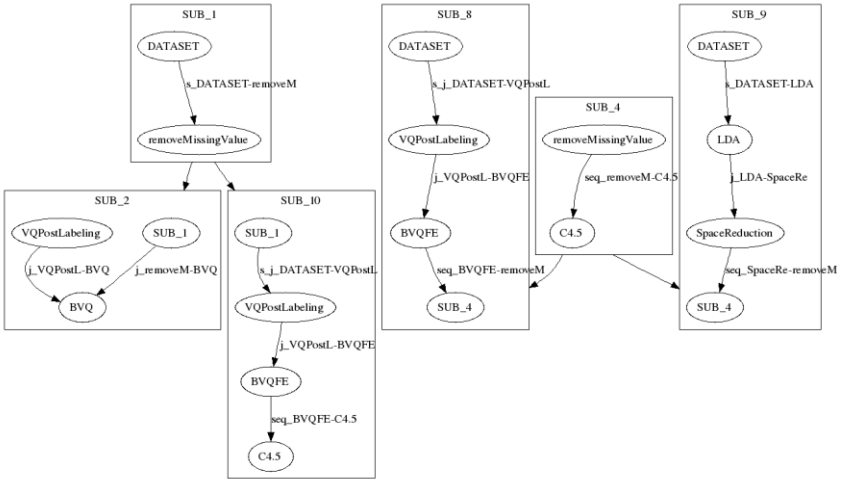

Fig. 6. First two levels of the lattice generated using C model

process for the application of SUBDUE algorithm have been discussed and evaluated. The evaluation on real business processes has been made difficult by the lack of a sufficient number of process schemas, hence we turned to a specialized domain like data mining, exploiting processes automatically generated by an ontology-based composer tool. Nevertheless, this activity allowed to gain useful insights on the method and on the particular domain as well. For instance, from the analysis of the generated lattice we were able to recognize typical patterns of the KDD methodology and we gained insights about some missing or wrong information in the ontology guiding the activity of process generation. The proposed method can find application in a variety of activities related to business process management: first, it can be exploited to individuate similarities and differences in the implementation of certain

processes at different companies, enlightening overlaps, complementarities and heterogeneities, hence supporting enterprise integration at the process level. Second, recurrent common substructures can be exploited to define reference prototype processes and best practices (or common bad practices). Third, the method can be exploited to organize a process repository to enhance search and retrieval. We plan to gather a sufficient number of business processes in order to concretely deal with these applications.

5 Bibliography

1. Jonyer, I., Cook, D. and Holder, L. (2001) Graph-Based Hierarchical Conceptual Clustering, in: Journal of Machine Learning Research, Vol. 2, pp. 19–43.
2. Ouvans, C., Dumas, M., ter Hofstede, A.H.M. and Van der Aalst, W.M.P. (2006) From BPMN Process Models to BPEL Web Services, 2006. International Conference on Web Services, pp.285-292, Chicago, IL, 18-22 Sept. 2006
3. Van der Aalst, W.M.P. and Weijters, A.J.M.M. (2004) Process mining: a research agenda, Computers in Industry 53:231–244.

Absorptive Capacity In Service Innovation: the Role of IT Capabilities

Luca Sabini, Paolo Spagnoletti

CeRSI-LUISS Guido Carli
{lsabini, pspagnoletti}@luiss.it

Abstract. This working paper sets up the basis for investigating the role of IT capabilities in the context of large and distributed service organizations. The idea grounded in this article resides in the possibility that absorptive capacity has an impact on service innovation. Addressing some theories on absorptive capacity and on service innovation we provide a brief insight on the importance that IT skills and capabilities can have on service innovation in large organization which are widespread distributed in different geographical location.

1 Introduction

The growing intensity and dynamism of competition across product markets has had profound implications for the evolution of strategic management. Increasing turbulence of the external business environment has focused attention upon the importance of learning fast how to behave within the markets and react to reach good performances (Grant, 1996). *Knowledge* and the processes through which knowledge is obtained and understood have emerged as the most strategically-significant resources of the firm.

In order to clarify the relationship between knowledge and firm performance/innovation, the concept of absorptive capacity has been introduced by Cohen and Levinthal (1990). Despite this construct has been developed focusing on product innovation in the manufacturing industry, some of its assumptions seems to apply also to service and open innovation domains. In fact, it allows explaining organizational phenomena such as the need to evaluate and incorporate externally generated technical knowledge into the firm which are even more relevant in the emerging open networked environments.

Open innovation refers to the environments in which multiple actors (both public and private) collaborate in delivering innovative services, each contributing with its own resources and capabilities, and where the underlying business models are attractive to all of the participants involved (Chesbrough 2003). In such collaborative environments, value is created via service innovation and by mediating between customer's needs, organizational resources and capabilities, financial arrangements,

and technological possibilities (Bouman and Fielt 2008, Chesbrough and Rosenbloom 2002).

In this context, information technology (IT) has been widely recognized as one of the firm capabilities which have a potential impact on the development of new products, services and the associated business processes (Swanson 1994, Swanson and Ramiller 2004). Nevertheless, the process by which IT based innovation is undertaken in cooperative service environments still needs further investigation.

In this working paper we address this topic by providing the theoretical basis for a further analysis of innovation processes in cooperative service environments. It lies on the constructs of absorptive capacity. In particular, the following research questions have been addressed: 1) to which extent absorptive capacity constructs apply to the service and open innovation contexts? 2) how do they relate to IT capabilities?

The paper is organized as follows. A theoretical background section will introduce the underpinning theories and concepts applied to a traditional organization. Then, a discussion session will briefly present the proposed framework.

2 Theoretical Background

The capability to learn is very important for an organization. The process through which it is achieved has been widely investigated in the management literature. This process has been divided into four main steps (Huber, 1991): *knowledge acquisition* (process by which knowledge is obtained), *information distribution* (process by which information from different sources is shared and thereby leads to new information understanding), *information interpretation* (process by which distributed information is given one or more common understood interpretations), *organizational memory* (means by which knowledge is stored for future use). All these elements concur to the building of organizational learning.

Information interpretation has been defined as “the process of translating events and developing shared understandings and conceptual schemes” (Daft & Weick, 1984). A particular aspects that Huber (1991) underlines is information overload. Interpretation within or across organizational units is less effective if the information to be interpreted exceeds the units' capacity to process the information adequately.

These concepts move to the idea that, due to the amount and the scale of information, there is no organizational learning when organizations tackle too many information.

The research of Levitt and March on the problem of interpretation of past experiences states that it is a process based on a small number of observations in a complex, changing organization. The events that happen are not always obvious, and the causality of events is difficult to untangle, the difference between success and failure of a given action is not always clear (Levitt & March, 1988).

The main problem with the learning theories described above is linked with the limit faced by an organization for allowing knowledge to flow inside the organization itself. Absorptive capacity is a limit to the rate of information that a firm can absorb.

The seminal article on absorptive capacity highlights the idea that the capacity of an organization to absorb external knowledge (recognize, evaluate, assimilate and apply) is a function of the level of prior related knowledge (Cohen and Levinthal, 1990). This assumption is very important because it stresses the importance to get ground knowledge of a particular subject in order to have the possibility of increasing organizational innovation by exploiting external one. For organizations this implication means that investing in "related" knowledge can be fruitful for their possibility to increase organizational innovation.

Other researches point out the different perspective that absorptive capacity (AC) has on *potential* and *realized* absorptive capacity. They highlight that while the ability to value and acquire external knowledge is a *potential* AC, the organization cannot gain positive outcome if this potential AC is not supported by *realized* AC which is a function of leveraging absorbed knowledge (Zahra and George, 2002). Therefore, the development of both these elements of AC is crucial for an organization.

The use of these concepts has been useful for understanding the possibility to move the absorptive capacity from the R&D department (Cohen and Levinthal, 1990), to more decentralized units such as local managers of subsidiaries. These people can easily catch ideas from external environment and formalize them in a way to suggest the production of new products and services to the organization. This process is subject to both the ability of organization and the ability of local managers. The former consists in providing managers with tools for accurately interpret their environment. The second ability lies in the capability of scanning the environment and identifying which type of new service/product can be implemented by organization.

3 Discussion

Previous research on the sources of innovation has demonstrated that organizational innovation results from borrowing rather than invention. Indeed, the ability to exploit external knowledge is a critical component of innovative capabilities. Furthermore, information originating from other internal units in the firm, outside the formal innovating unit (i.e., the R&D lab), such as marketing and manufacturing, has also received attention in the product industry (Cohen and Levinthal, 1990).

In large service organizations (i.e. banks) which are widespread distributed within different geographical environments and where subsidiaries encounter different customer needs and a variety of possible partners, the role of local managers and of integration mechanisms become crucial. In fact, on the one hand local managers represent the gatekeepers which can support organizations in the definition of competitive strategies and in the development of new services. They have the possibility to acquire, assimilate, transform and exploit knowledge in order to achieve organizational innovation. Thus, local managers should possess a set of skills allowing them to recognize the potential value of a business idea and to communicate it to other organizational units. On the other hand, integration mechanisms (i.e. specialized actors, supporting tools) are needed in order to enable the transfer of

knowledge from local managers to the organizational units in charge of implementing new services.

Therefore, the effectiveness of the overall innovation process lies on both the quality of the interpretation that managers perform on the environmental needs and opportunities (*potential* absorptive capacity) and the effectiveness of the integration mechanisms which exploit the service innovation (*realized* absorptive capacity).

When services are heavily supported by IT, interoperability represents a prerequisite of open innovation environments and the above mentioned "*prior related knowledge*" must necessarily refer also to IT skills and capabilities.

As a result of this conceptual analysis on the applicability of absorptive capacity concepts to a cooperative service environment, the following research questions arise: 1) which type of IT skills and capabilities (*prior related knowledge*) positively affect the *potential* absorptive capacity of local managers? 2) which integration mechanisms can be implemented at organizational/inter-organizational level in order to positively affect the *realized* absorptive capacity and to increase the efficiency of assimilation and transformation?

These research questions should be further empirically investigated in order to gain insights on the role of IT skills and capabilities in the organizational innovation processes.

4 References

Chesbrough HW (2003) Open Innovation: the new imperative for creating and profiting from technology. *Harvard Business School Press*

Chesbrough H & Rosenbloom RS (2002) The role of the business model in capturing value from innovation: evidence from Xerox. *Industrial and Corporate Change* 11(3): 529-555.

Cohen, W. M., & Levinthal, D. (1990). Absorptive Capacity : A New Perspective on Learning and Innovation. *Administrative Science Quarterly*, 35(1), 128-152.

Daft, R., & Weick, K. (1984). Toward a model of organizations as interpretation systems. *Academy of management review*.

Grant, R. M. (1996). Prospering in Dynamically-Competitive Environments: Organizational Capability as Knowledge Integration. *Organization Science*, 7(4), 375-387.

Huber, G. P. (1991). Organizational Learning: The Contributing Processes and the Literatures. *Organization Science*, 2(1), 88-115.

Lane, P.J., Koka, B.R., and Pathak, S. 2006. "The Reification of Absorptive Capacity: A Critical Review and Rejuvenation of the Construct," *Academy of Management Review* (31:4), pp. 833-863

Levitt, B., & March, J. (1988). Organizational Learning. *Annual Reviews*, 14, 319-340.

Swanson, E. (1994). Information systems innovation among organizations. *Management Science*, 40(9).

Swanson, E., & Ramiller, N. (2004). Innovating mindfully with information technology. *MIS Quarterly*, 28(4), 553-583

Zahra, A. S., & George, G. (2002). Absorptive capacity: a review, reconceptualization, and extension. *Academy of Management Review*, 27(2), 185-203.